PENGUJIAN SISTEM ENKRIPSI-DEKRIPSI DENGAN METODE RSA UNTUK PENGAMANAN DOKUMEN

SUPRIYONO
Sekolah Tinggi Teknologi Nuklir – BATAN
Jl. Babarsari Kotak Pos 6101/YKBB Yogyakarta.
Email : masprie_sttn@yahoo.com

Abstrak

Telah diuji suatu model pengamanan dokumen yang dapat digunakan sebagai salah satu instrumen sistem pengamanan dokumen khususnya untuk dokumen teks. Adapun prinsip pengamanan dokumen ini adalah bagaimana sistem dapat mengamankan proses penyimpanan dan pengiriman dokumen. Mula-mula dokumen dalam bentuk teks dienkripsi. Sehingga dokumen tidak dapat dibaca oleh siapapun. Karena teks telah berubah menjadi susunan huruf yang teracak. Dokumen yang susunan hurufnya telah teracak tersebut jika ingin dibaca oleh pemilik dokumen, maka dokumen tersebut harus dibuka dengan dekripsi. Dalam penelitian ini, metode yang digunakan adalah metode RSA, dimana metode tersebut menggunakan perhitungan matematika yang rumit dan disertai dengan kunci pengaman awal (dengan private key maupun dengan public key) sehingga amat sulit untuk ditembus oleh hacker. Sistem ini dibangun dengan perangkat lunak Borland Delphi 7. Hasil pengujian ini menunjukkan bahwa sistem dapat menyimpan dan mengirimkan dokumen baik pengiriman melalui internet maupun intranet dalam bentuk susunan huruf yang terenkripsi dan mengembalikan ke bentuk dokumen semula dengan cara dekripsi. Dilakukan pula pengujian proses enkripsi dan dekripsi untuk dokumen dengan ukuran memori yang bermacam-macam.

Kata Kunci : Pengujian, Dokumen, Enkripsi, Dekripsi, RSA, Waktu proses.

Abstract

A model of document protection which was tested as one of the instruments, especially text document. The principle of the document protection was how the system was able to protect the document storage and transfer processes. Firstly, the text-formed document was encrypted; therefore, the document cannot be read for the text was transformed into random letters. The letter-randomized text was then unfolded by the description in order that the document owner was able to read it. In the recent research, the method adopted was RSA method, in which it used complicated mathematics calculation and equipped with initial protection key (with either private key or public key), thus, it was more difficult to be attacked by hackers. The system was developed by using the software of Borland Delphi 7. The results indicated that the system was capable to save and transfer the document, both via internet and intranet in the form of encrypted letter and put it back to the initial form

of document by way of description. The research also tested for encrypted and decrypted process for various memory size documents.

Keywords : Testing, Document, Encryption, Decryption, RSA, Proses time.

PENDAHULUAN

Suatu organisasi pasti ada suatu dokumen yang bersifat rahasia. Dokumen yang bersifat rahasia tersebut perlu dibuatkan sistem penyimpanan dan pengirimannya agar tidak terbaca oleh orang-orang yang tidak bertanggung jawab. Dokumen yang perlu diamankan adalah dokumen-dokumen penting dan bersifat rahasia, baik dokumen tersebut tersimpan sebagai file di dalam komputer pribadi maupun file yang dikirim lewat email. Untuk menyimpan dokumen tersebut agar benar-benar aman, tentunya dilakukan sistem pengamanan yang baik, yang bebas dari jangkauan penjahat atau pada orang-orang yang tidak berhak. Baik bebas jangkauan secara phisik maupun secara sistem. Untuk bebas secara phisik, maka faktor orang sebagai penjaga memegang peranan yang penting, sedangkan aman menurut sistem adalah dokumen tersebut tersimpan dalam kondisi tidak dapat dibaca oleh orang.

Untuk membangun sistem penyimpanan dokumen yang hasil simpanannya tidak dapat dibaca oleh orang, dalam penelitian ini telah dikembangkan model sistem pengamanan dengan proses enkripsi dan dekripsi dengan metode RSA. Prosedur metode RSA ini, dalam proses menggunakan perhitungan matematika yang rumit dan disertai dengan kunci pengaman awal (dengan private key maupun dengan public key) sehingga amat sulit untuk ditembus oleh hacker[2,3]. Dalam penelitian ini karena keterbatasan penulis untuk mendapatkan dokumen nuklir yang sebenarnya, maka dokumen nuklir tersebut sebagai obyek penelitian hanya berupa simulasi atau perumpamaan dan sistem ini dapat digunakan hanya untuk dokumen berupa teks, bukan berupa gambar atau suara.

Untuk membangun sistem ini digunakan perangkat lunak Borland Delphi dengan sistem operasi Windows XP dan dapat dijalankan dengan komputer Pentium IV atau di atasnya. Sistem dibangun dengan kaidah-kaidah teknik informatika, yaitu dengan diawali analisis kebutuhan sistem dan diakhiri dengan pengujian sistem.

Hasil pengujian menunjukkan bahwa bahwa sistem dapat menyimpan dan mengirimkan dokumen baik pengiriman melalui internet maupun intranet dalam bentuk susunan huruf yang terenkripsi dan mengembalikan ke bentuk dokumen semula dengan cara dekripsi. Telah dilakukan pula pengujian

proses enkripsi dan dekripsi untuk dokumen dengan ukuran memori yang bermacam-macam.

Landasan Teori

Cryptography salah satu ilmu yang sangat penting dan mulai menonjol sejak digunakan pada aplikasi radio dan pengiriman perintah perang pada Perang Dunia ke II. Aplikasi ini pada PD II digunakan untuk membuat kode dan membakar atau mengacak kode lawan. Saat ini cryptography semakin dikembangkan untuk aplikasi-aplikasi pertahanan keamanan maupun untuk aplikasi bisnis.

Pada intinya sistem pengamanan dokumen dengan cryptography ada 2 langkah[1,3], yaitu :

1. Proses Enkripsi
   Proses enkripsi adalah suatu proses yang mengubah plainteks (kode sesungguhnya) menjadi ciperteks (kode rahasia). Untuk merubah plainteks ke ciperteks digunakan fungsi matematika dan kunci.
2. Proses Dekripsi
   Proses dekripsi adalah suatu proses yang mengubah ciperteks menjadi plainteks, dimana dokumen yang sudah teracak dikembalikan ke dokumen semula yang juga menggunakan fungsi matematika dan kunci.

Sebelum proses Proses enkripsi maupun dekripsi dilakukan, ada satu pengaman awal yaitu menentukan kunci sandi kunci pengaman (*key pairs*) yang terdiri dari *private key*, *public key* dan *modulo* yang digunakan untuk membuka dan mengunci system. Setelah system dapat dibuka dengan kunci pengaman, proses enkripsi maupun dekripsi dapat dilakukan, baik dilakukan dengan proses enkripsi dan enkripsi sekali saja maupun proses enkripsi dan dekripsi yang dilakukan berkali-kali agar semakin terjamin kerahasiaannnya. Banyak metode yang berasal dari fungsi matematika yang digunakan untuk proses enkripsi maupun dekripsi. Salah satu metode tersebut yang digunakan dalam penelitian ini adalah metode RSA[1]. Pada metode RSA yang berperan penting adalah penyandian blok, yaitu setiap proses perhitungan enkripsi dan dekripsi dilakukan dengan hitungan per blok.

Metode RSA digagas oleh Ron Rivest, Adi Shamir dan Leonard Adleman dari MIT tahun 1977[2]. Walaupun metode RSA sudah berumur 30 tahun, tetapi metode ini termasuk metode pengaman dokumen yang cukup handal.

Adapun rumus matematika beserta prosedur metode RSA adalah sebagai berikut :

1. Ambil secara random dua bilangan prima p dan q yang besar dan berbeda, namun ukuran keduanya (jumlah digit dalam basis bilangan yang digunakan) harus sama.
2. Hitung modulus n dan fungsi Euler's Totient $\phi(n)$ dengan rumus :

$$n = p\ q \tag{1.a}$$

$$\phi(n)=(p-1)[q-1] \tag{1.b}$$

dengan :
n = modulus (*public key*)
p dan q = dua bilangan prima yang dimunculkan secara random.

3. Pilih suatu bilangan integer *e* sedemikian hingga

$$I<e<\phi(n) \text{ dan } gcd(e,\phi(n))=1 \tag{2}$$

dengan :
I = bilangan integer
e = *public key* (kunci enkripsi)
gcd = persekutuan pembagi terbesar (*greatest common divisor*)

4. Hitung nilai integer d dimana $I < d < \phi(n)$ sedemikian hingga

$$d = e^{-1} \bmod \phi(n) \text{ atau } ed = I(\bmod \phi(n)) \tag{3}$$

dengan :
d = *private key* (kunci dekripsi)

5. Membangun tabel untuk mempresentasikan tiap karakter.
6. Plainteks (teks yang akan dienkripsi) disandikan dengan angka-angka sesua dengan tabel yang terbentuk oleh proses 5 dan akan diperoleh suatu nilai *M* yang merupakan kumpulan angka-angka dari plaintext, kemudian kumpulan angka-angka tersebut diblok tiap 4 angka menjadi $m_1, m_2, \cdots, m_n$. Proses enkripsi dilakukan per blok dan masing-masing blok rumus enkripsinya adalah $c_1 = m_1^{\,e}(\bmod n)$, $c_2 = m_2^{\,e}(\bmod n)$, …..dst, sehingga menghasilkan nilai *C* dimana *C* merupakan kumpulan angka-angka dari $c_1, c_2, \cdots, c_n$.

7. Proses dekripsi dilakukan dengan menggunakan logika seperti langkah 6 dengan melakukan perhitungan terbalik, yaitu $m_1 = c_1^{\,d} (\text{modn})$, $m_2 = c_2^{\,d} (\text{modn})$,dst, sehingga menghasilkan nilai M dimana M = m_1 m_2 m_3 Nilai akhir M tersebut dipresentasikan balik dengan table yang dibangun seperti pada proses 5 di atas.

METODE PENELITIAN

Sesuai dengan kebutuhan suatu sistem informasi[4] dalam membangun perangkat lunak diperlukan langkah-langkah sebagai berikut :

Merancang Kunci Pengaman

Sistem agar betul-betul aman, maka seseorang sebelum membuka sistem wajib mengisikan kunci pengaman berupa sandi *private key*, *public key* dan *modulo* yang dibangkitkan secara random. Sandi kunci pengaman tersebut juga dapat diubah sewaktu-waktu sesuai dengan kepentingan pemilik sistem. Artinya jika sandi kunci pengaman oleh pemilik sistem dirasa sudah tidak aman, maka pemilik sistem dapat segera merubahnya. Tampilan kunci pengaman ini terdiri dari :

1. *Generate Key Pair* merupakan bottom yang berfungsi untuk membangkitkan sandi *Private Key*, *Public Key* dan *Modulo* dan ada fasilitas *save* untuk menyimpan sandi-sandi tersebut dalam sebuah nama file..
2. *Delete key pair* merupakan bottom yang berfungsi untuk menghapus sandi kunci pengaman.
3. *Close* merupakan bottom yang digunakan jika sudah selesai membuat sandi kunci pengaman.

Merancang Prosedur Enkripsi dan Dekripsi

Setelah rancangan pembuatan kunci pengaman selesai dilakukan, langkah selanjutnya adalah merancang proses enkripsi dan dekripsi, yaitu berupa prosedur dengan dasar pembuatannya mengacu langkah-langkah pada bab landasan teori. Pada proses enkripsi inputnya merupakan teks yang berasal dari semua karakter yang ada di file ASCII, yaitu :

1. *Plain Text* merupakan area yang berfungsi sebagai input dokumen yang akan dirahasiakan.
2. *Key Pair* merupakan area inputan untuk mengisi kunci sandi pengaman.
3. *Chiper Text* merupakan area tampilan hasil enkripsi, dimana tampilannya sudah tidak dapat dibaca oleh orang yang tidak punya alat dekripsinya. Karena berupa barisan angka-angka yang manusia tidak akan pernah tahu maknanya.

4. *Encrypt* merupakan bottom untuk merubah dokumen asli menjadi dokumen yang sudah terenkripsi.

Sedangkan pada proses dekripsi tampilannya hampir sama dengan tampilan proses enkripsi, hanya fungsi *Plain Text* dan *Chiper Text* terbalik, yaitu area *Plain Text* diisi oleh hasil enkripsi dan area *Cipher Text* merupakan hasil dekripsi yang notabene merupakan dokumen asli (kembali ke proses dokumen asli).

Merancang Antar Muka (Interface) Menu Utama, Tampilan Membuat Kunci Pengaman Tampilan Proses Enkripsi Maupun Proses Dekripsi

Rancangan tampilan antar muka terdiri dari :
1. Antar Muka Menu Utama.
2. Antar Muka Pembangkitan Sandi Kunci Pengaman.
3. Antar Muka Pengisian *Private Key*, *Public Key* dan *Modulo*.
4. Antar Muka Proses Enkripsi.
5. Antar Muka Proses Dekripsi.

Membuat Program

Program komputer yang digunakan untuk membangun sistem ini adalah perangkat lunak Borland Delphi 6 dengan alasan bahwa Borland Delphi 6 merupakan bahasa komputasi teknis yang sangat populer dan sangat mudah digunakan serta mudah pula untuk dipahami struktur bahasanya. Selain itu, Borland Delphi 6 mempunyai area besar, mempunyai type data baru, struktur dan fasilitas-fasilitas grafik yang lebih baik dan cepat pemrosesannya. Dengan listing program yang sangat panjang, maka listing program dalam penelitian ini tidak dapat ditampilkan dalam makalah ini. Sedangkan prosedur proses enkripsi dan dekripsi ditampilkan pada lampiran.

Pengujian Sistem

Setelah sistem selesai dibangun, maka harus diuji apakah sistem dapat berjalan dengan baik dan mudah dioperasikan. Pengujian dilakukan dengan melakukan langkah – langkah :
1. Mengisi kunci pengaman dengan membangkitkan kunci random dan mengisi kata kunci.
2. Menuliskan dokumen yang akan disimpan pada menu *Encrypt.*
3. Menuliskan hasil enkripsi untuk diubah menjadi dokumen semula pada menu *Dekripsi.*

Secara detail pengujian program disampaikan pada bab Hasil dan Pembahasan berikut ini.

HASIL DAN PEMBAHASAN

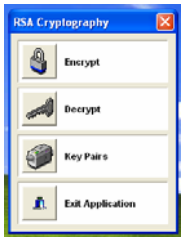
Dimisalkan ada untuk 2 orang yang pucuk pimpinan yang ingin berkomunikasi lewat email atau lewat media transfer data lainnya, maka kedua pimpinan tersebut masing-masing harus mempunyai sistem pengamanan dokumen (mesin ekskripsi dan dekripsi) ini dan mempunyai kunci pembuka yang sama. Sebagai contoh pengamanan dokumen pada suatu organisasi yang sedang melakukan persiapan pembangunan PLTN dan dilanjutkan dengan pengoperasiannya :

Pada Saat Persiapan Pembangunan PLTN

Dimisalkan pimpinan A ingin mengirim informasi yang bersifat rahasia kepada pimpinan B, dengan kalimat “Urutan Penawaran PLTN dari vendor : Mitsubishi Japan 30 triliyun rupiah, Westinghouse Amerika 29,1 triliyun rupiah dan Siemens Jerman 28,6 triliyun rupiah.”, maka langkah-langkah yang dilakukan :

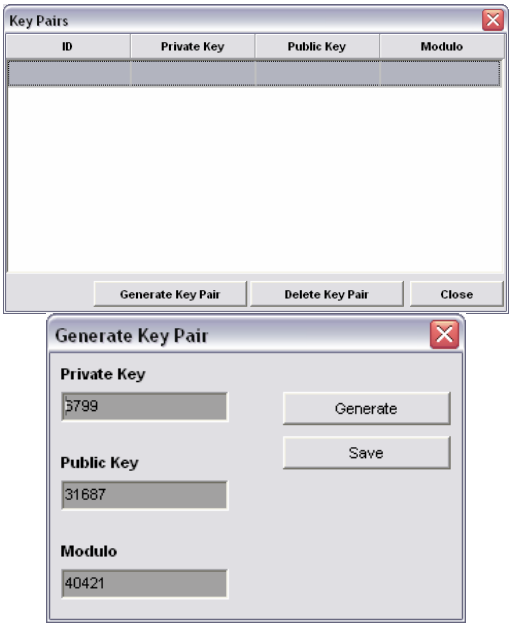
Yang dilakukan Pimpinan A

Membuat kunci pembuka, yaitu dengan memilih tombol Key Pairs pada Gambar 1 di bawah ini.



Gambar 1. Menu Utama Sistem Pengamanan Dokumen Nuklir

Hasilnya adalah tampilan seperti Gambar 2 berikut :

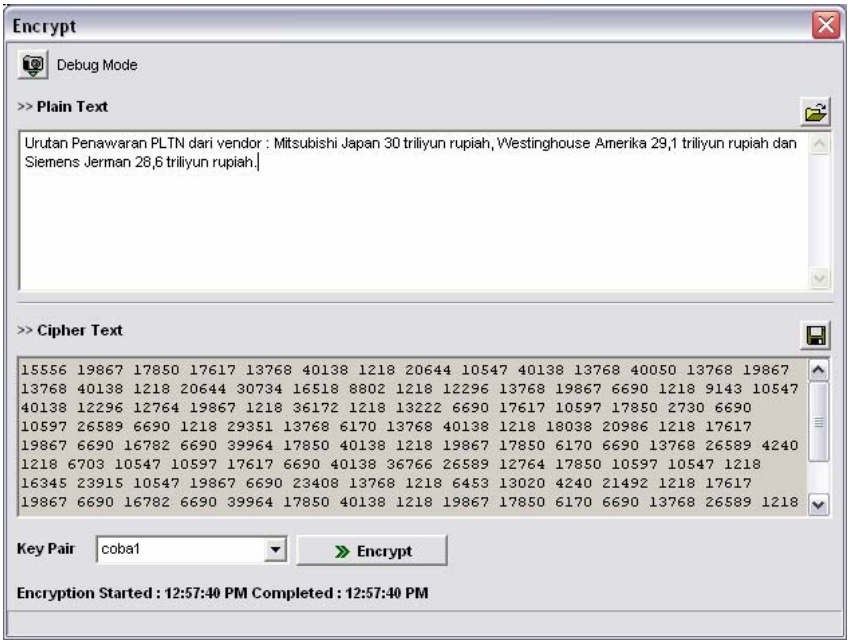


Gambar 2. Menu Pembuatan Sandi Kunci Pengaman

Dari Gambar 2 di atas menunjukkan bahwa Pimpinan A mempunyai sandi kunci pengaman, yaitu : *Private Key* = 6799, *Public Key* = 31687 dan *Modulo* = 40421. Sandi kunci pengman itulah yang harus dikirim ke pimpinan B.

1. Mengisi kalimat yang akan dirahasiakan di atas di area *Plain Text* pada jendela enkripsi.

   Setelah memilih tombol Encrypt pada Gambar 1, copykan kalimat yang akan dirahasiakan di atas ke dalam area *Plain Text* dan hasil enkripsinya adalah barisan angka pada area *Cipher Text* seperti tampilan pada Gambar 3 berikut :

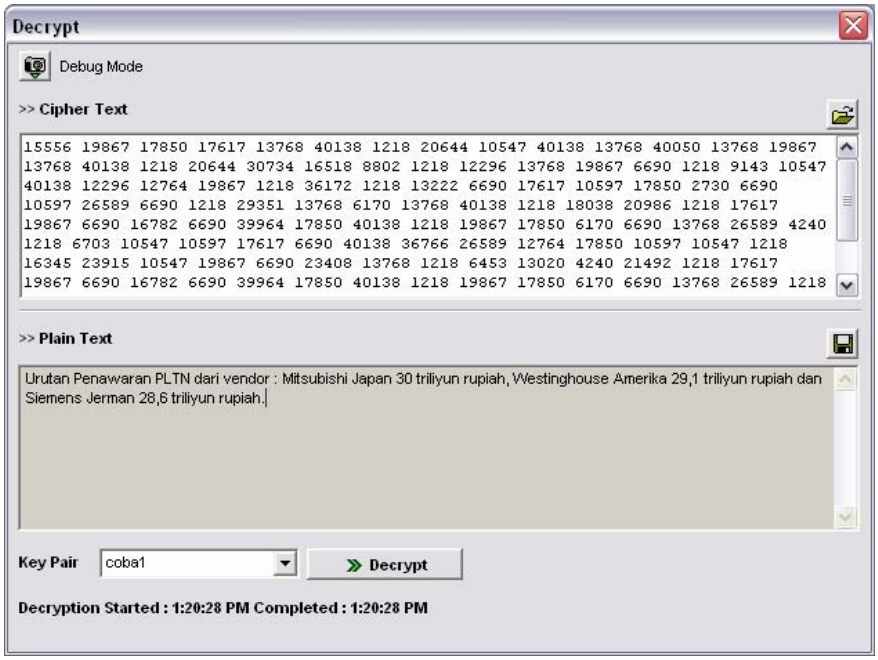


Gambar 3. Tampilan Dokumen Asli dan Yang Sudah Terenkripsi

2. Mengirim sandi kunci pengaman dengan sandi kunci pengaman *Private Key* = 6799, *Public Key* = 31687 dan *Modulo* = 40421 ke Pimpinan B.
3. Mengirim kalimat yang dirahasikan berupa barisan angka-angka yang tertampil pada area *Cipher Text* seperti yang tertampil pada Gambar 3 di atas.

Yang Dilakukan Pimpinan B

1. Membuka Mesin enkripsi dan dekripsi dengan sandi kunci pengaman sama dengan sandi kunci pengaman milik pimpinan A, yaitu : *Private Key* = 6799, *Public Key* = 31687 dan *Modulo* = 40421.
2. Mengisi dokumen yang sudah terenkripsi oleh pimpinan A kedalam mesin dekripsi. Hasil dekripsinya ditampilkan seperti Gambar 4 berikut :

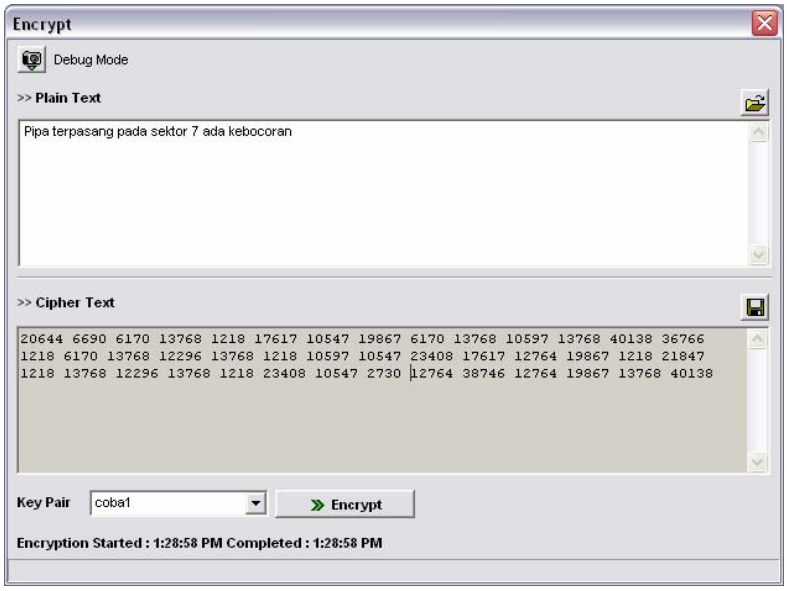


Gambar 4. Hasil Dekripsi

3. Hasilnya adalah kalimat ” Urutan Penawaran PLTN dari vendor : Mitsubishi Japan 30 triliyun rupiah, Westinghouse Amerika 29,1 triliyun rupiah dan Siemens Jerman 28,6 triliyun rupiah. Persis seperti dokumen rahasia milik pimpinan A di atas.

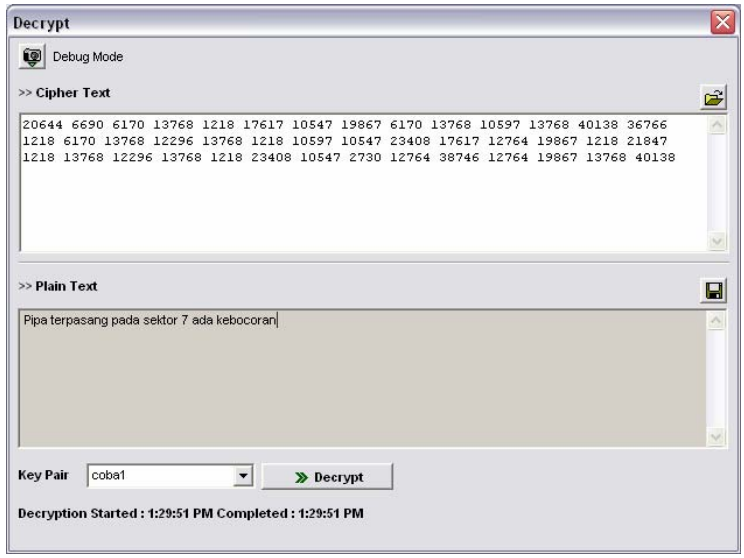
Pada Saat Pembangunan PLTN

Mesin pengamanan dokumen ini juga dapat digunakan pada saat pembangunan PLTN, Misalnya pada saat pembangunan PLTN ada masalah dengan dokumen yang kalimatnya adalah “Pipa terpasang pada sektor 7 ada kebocoran”, maka proses enkripsi dan dekripsinya adalah :



Gambar 5. Proses Enkripsi Pada Saat Pembangunan PLTN

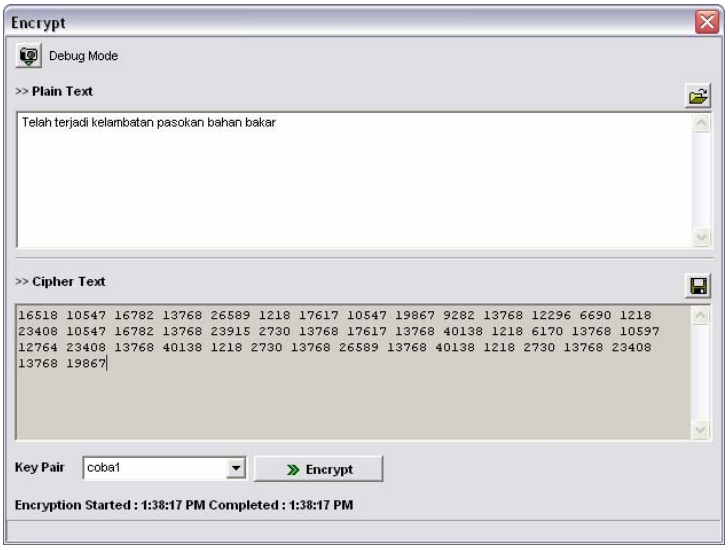
Dan hasil dekripsinya adalah :



Gambar 6. Proses Dekripsi pada saat Pembangunan PLTN

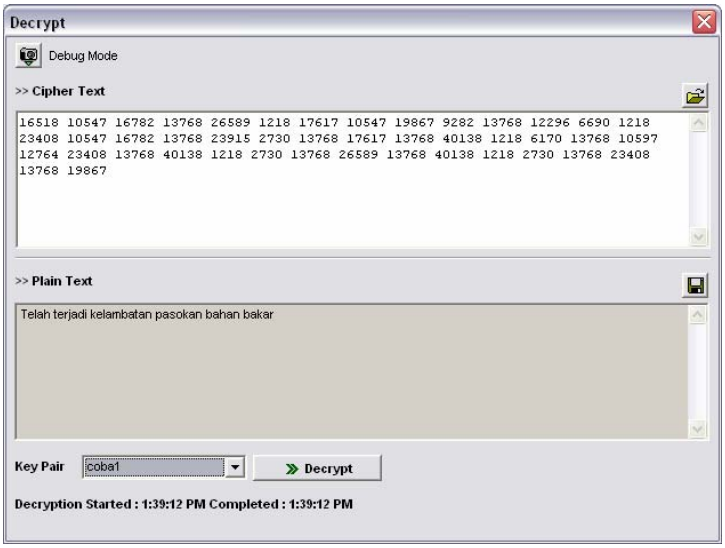
Pada Saat Operasi PLTN.

Demikian juga pada saat operasi PLTN, misanya ada kalimat “Telah terjadi kelambatan pasokan bahan bakar”, maka hasil enskripsi dan dekripsinya dapat ditampilkan pada Gambar 7 dan Gambar 8 berikut :



Gambar 7. Proses Enkripsi pada saat Operasi PLTN

Dan hasil dekripsinya adalah :



Gambar 8. Proses Dekripsi pada saat Operasi PLTN

Proses enkripsi dan dekripsi pada mesin pengamanan ini juga dapat dilakukan bertingkat-tingkat jika menginginkan dokumen menjadi super rahasia. Misalnya sebuah kalimat ”PLTN di Jepara”, maka proses enkripsi dan dekripsi bertingkat dua menghasilkan bentuk dokumen seperti tabel berikut :

Tabel 1. Proses Enkripsi dan Dekripsi Dua Tingkat

Pemimpin A		Pemimpin B	
Dokumen Awal	PLTN di Jepara	Dokumen dari Pemimpin A	6453 20986 30931 4286 4286 1218 18038 20986 21847 18038 4286 1218 21492 30931 315 21492 4599 1218 4599 4599 20986 6453 1218 21492 6453 21492 4599 1218 21492 6453 6453 13020 30931 1218 30931 30931 13020 20986 1218 21492 6453 21492 4599 1218 6453 13020 18038 315 21492 1218 21492 20986 315 4286 21847 1218 30931 21492 21847 20986 1218 21492 18038 21847 30931 4599 1218 21492 13020 4599 30931 21847 1218 21492 18038 21847 30931 4599
Hasil Enkripsi I	20644 30734 16518 8802 1218 12296 6690 1218 29351 10547 6170 13768 19867 13768	Hasil Dekripsi I	20644 30734 16518 8802 1218 12296 6690 1218 29351 10547 6170 13768 19867 13768
Hasil Enkripsi II	6453 20986 30931 4286 4286 1218 18038 20986 21847 18038 4286 1218 21492 30931 315 21492 4599 1218 4599 4599 20986 6453 1218 21492 6453 21492 4599 1218 21492 6453 6453 13020 30931 1218 30931 30931 13020 20986 1218 21492 6453 21492 4599 1218 6453 13020 18038 315 21492 1218 21492 20986 315 4286 21847 1218 30931 21492 21847 20986 1218 21492 18038 21847 30931 4599 1218 21492 13020 4599 30931 21847 1218 21492 18038 21847 30931 4599	Hasil Dekripsi II	PLTN di Jepara

Model pengamanan dokumen ini dapat juga dilakukan untuk proses enkripsi dan dekripsi lebih dari dua tingkat.

Dalam penelitian ini telah diuji untuk teks yang 40 kilobytes, waktu yang digunakan untuk proses enkripsi sekitar 40 detik, sedangkan untuk proses dekripsinya sekitar 60 detik. Pada dasarnya sistem ini jika masih berupa teks, untuk ukuran berapapapun tidak ada masalah, tetapi sebetulnya untuk dokumen-dokumen rahasia biasanya tidak terlalu besar ukurannya. Beberapa hasil pengujian waktu proses enkripsi dan dekripsi ditampilkan pada Tabel 2 berikut :

Tabel 2. Waktu Enkripsi dan Dekripsi untuk Dokumen Teks.

Besar Memori Dokumen	Waktu Enkripsi	Waktu Dekripsi
40 kbyte	40 detik	60 detik
60 kbyte	62 detik	87 detik
100 kbyte	114 detik	132 detik
150 kbyte	178 detik	198 detik
200 kbyte	235 detik	247 detik

KESIMPULAN

Dari hasil penelitian ini, kesimpulannya adalah sebagai berikut :

1. Metode RSA dengan basis perhitungan matematika keamanannya dapat dipertanggung jawabkan keakurasiannya, karena dapat dilakukan proses enkripsi dan dekripsi bertingkat.
2. Dengan pengamanan ganda, yaitu dengan adanya Key Pair, sistem ini baik untuk sistem stand alone maupun untuk sistem jaringan.
3. Sistem ini dapat diaplikasikan untuk teks-teks lain untuk ukuran yang besar maupun yang kecil.
4. Semakin besar memori dokemen, waktu enkripsi dan dekripsi juga semakin besar.

DAFTAR PUSTAKA

1. MENEZES.A.J., at. All., 1996, *Handbook of Applied Cryptography*, CRC Press, London.
2. http://id.wikipedia.org/Wiki/RSA#Sejarah_RSA. Diunduh 10 Agustus 2008.
3. DOROTHY,R. AND ELIZABETH,D.., 1993, “Cryptography and Data Security”, Addison-Wesley Publishing, Comp.
4. JOGIYANTO,H.M., 1990, *Analisis dan Desain Sistem Informasi*, Andi Offset, Yogyakarta.

5. JOHNSONBAUGH,R., 1998, *Matematika Diskrit*, Prenhallindo, Jakarta.

LAMPIRAN

Prosedur Modul Program Enkripsi

```
procedure Tform_process.EncryptText(public_key, modulo : integer);
var plaintxt, ciphertxt, status : string;
    cipher : integer;
    i, strlen, progress : integer;
begin
  plaintxt:= memo_top.Lines.Text;
  ciphertxt:= '';
  doDebug('>>Encrypting '+inttostr(length(plaintxt))+' characters');
  pbar.Position:= 0;
  status:= 'Encryption Started : '+TimeToStr(Time);
  strlen:= length(plaintxt);
  for i:= 1 to strlen do
    begin
      doDebug('- Character #'+inttostr(i)+' : ');
      doDebug('  Plain  = '''+plaintxt[i]+''' ('+inttostr(Ord(plaintxt[i]))+')');
      cipher:= RSAEncrypt(Ord(plaintxt[i]),public_key, modulo);
      if ciphertxt<>'' then
        ciphertxt:= ciphertxt+' '+IntToStr(cipher)
      else
        ciphertxt:= IntToStr(cipher);
      doDebug('  Cipher = '+IntToStr(cipher));
      progress:= round(i/strlen*100);
      pbar.Position:= progress;
    end;
  pbar.Position:= 0;
  status:= status + ' Completed : '+TimeToStr(Time);
  status_label.Caption:= status;
  memo_bottom.Clear;
  memo_bottom.Lines.Text:= ciphertxt;
end;
```

Prosedur Modul Program Dekripsi

```
procedure Tform_process.DecryptText(private_key, modulo: integer);
var plaintxt, ciphertxt, status : string;
    cipher, plain : integer;
    i1, i2, idx : integer;
begin
  ciphertxt:= memo_top.Lines.Text;
  plaintxt:= '';
  i1:= 1;
  i2:= 1;
  idx:= 1;
  doDebug('>>Decrypting '+inttostr(length(ciphertxt))+' characters');
  status:= 'Decryption Started : '+TimeToStr(Time);
  while i2<length(ciphertxt) do
    begin
      if ciphertxt[i2]=' ' then
        begin
          cipher:= strtoint(Copy(ciphertxt,i1,i2-i1));
          doDebug('- Character #'+inttostr(idx)+' : ');
          doDebug('  Cipher = '+inttostr(cipher));
          plain:= RSADecrypt(cipher,private_key,modulo);
          plaintxt:= plaintxt+Char(plain);
          i1:= i2+1;
          doDebug('  Plain  = '+QuotedStr(Char(plain)));
          inc(idx);
        end;
      inc(i2);
    end;
  cipher:= strtoint(Copy(ciphertxt,i1,i2-i1+1));
  plain:= RSADecrypt(cipher,private_key,modulo);
  plaintxt:= plaintxt+Char(plain);
  status:= status + ' Completed : '+TimeToStr(Time);
  status_label.Caption:= status;
  memo_bottom.Clear;
  memo_bottom.Lines.Text:= plaintxt;
end;
```